Enterprise Linux Best Practices

Securing RHEL 5 Installations at McGill

McGill University Information Security

Konstantin Ryabitsev

Enterprise Linux Best Practices
Securing RHEL 5 Installations at McGill
Edition 1.2

Author Konstantin Ryabitsev *konstantin.ryabitsev@mcgill.ca*

A set of recommendations and general practices to follow in order to secure on-campus installations of Red Hat Enterprise Linux (RHEL) version 5.

Preface

1. Warnings

- Do not attempt to implement any of the recommendations in this guide without first testing them in a non-production environment.
- This document is only a guide containing recommended security settings. It is not meant to replace well-structured policy or sound judgment.
- This guide is tailored to Red Hat Enterprise Linux (*RHEL*) 5 and may not be applicable to earlier or later versions of RHEL.

2. Sources

This guide heavily borrows from the following documents:

- NSA's *Guide to the Secure Configuration of Red Hat Enterprise Linux 5*[1]
- SANS Press "Securing Linux: A Survival Guide for Linux Security"

3. Feedback

Please send corrections to *`infosec@mcgill.ca`* .

Introduction

The purpose of this guide is to help campus systems administrators to secure their installations of RHEL 5 so that they adhere to the guidelines put forth by McGill University Information Security. This guide is by no means exhaustive — each system is unique and must be reviewed and audited separately. However, the aim is to provide the first stepping stone toward increased security among Linux systems on campus.

The guide is intended for systems administrators, and assumes that its readers already possess broad knowledge of Red Hat Enterprise Linux. If you are not familiar with RHEL systems administration, then you should read documentation provided by Red Hat on *their website*[1].

Some instructions within this guide are complex. All directions should be followed completely and with understanding of their effects in order to avoid serious adverse effects on the system and its security.

1.1. General principles

The following general principles motivate much of the advice in this guide and should also influence any configuration decisions that are not explicitly covered.

1.1.1. Encrypt transmitted data whenever possible

Data transmitted over a network, whether wired or wireless, is susceptible to passive monitoring. Whenever practical solutions for encrypting such data exist, they should be applied. Even if data is expected to be transmitted only over a local network, it should still be encrypted. Encrypting authentication data, such as passwords, is particularly important.

1.1.2. Minimize software to minimize vulnerability

The simplest way to avoid vulnerabilities in software is to avoid installing that software. On RHEL, the RPM Package Manager allows for careful management of the set of software packages installed on a system. Installed software contributes to system vulnerability in several ways:

- Packages that include setuid programs may provide local attackers a potential path to privilege escalation.
- Packages that include network services may give this opportunity to network-based attackers.
- Packages that include programs which are predictably executed by local users (e.g. after graphical login) may provide opportunities for trojan horses or other attack code to be run undetected.

The number of software packages installed on a system can almost always be significantly pruned to include only the software for which there is an environmental or operational need.

[1] http://www.redhat.com/

1.1.3. Run different network services on separate systems

Whenever possible, a server should be dedicated to serving exactly one network service. This limits the number of other services that can be compromised in the event that an attacker is able to successfully exploit a software flaw in one network service.

1.1.4. Configure security tools to improve system robustness

Several tools exist which can be effectively used to improve a system's resistance to and detection of unknown attacks. These tools can improve robustness against attack at the cost of relatively little configuration effort. In particular, this guide recommends and discusses the use of IPTables for host-based firewalling, SELinux for protection against vulnerable services, and a logging and auditing infrastructure for detection of problems.

1.1.5. Defense in depth

Systems fail and people fail, including carefully constructed security defense lines and procedures. One of the general guidelines when it comes to security is to build in redundancy into your configuration, in case an attacker manages to bypass one of the outlying lines of defense. For example, if network ACLs failed to apply, your system should have local iptables configured as a fallback protection. If iptables happened to fail to start, your services should be configured to only accept connections from trusted sources. If an intruder managed to sneak in a sniffer onto a local fully-switched network, your traffic is encrypted point-to-point to prevent sniffing anyway.

1.2. How to use this guide

Readers should keep in mind the following things about this guide.

1.2.1. Test in non-production environment

This guidance should always be tested in a non-production environment before deployment. This test environment should simulate the setup in which the system will be deployed as closely as possible.

1.2.2. Root shell assumed

Most of the actions listed in this document are written with the assumption that they will be executed by the root user running the **`/bin/bash`** shell. Any commands preceded with a hash mark (**`#`**) assume that the administrator will execute the commands as root, i.e. apply the command via **`sudo`** whenever possible, or use **`su`** to gain root privileges if **`sudo`** cannot be used (e.g. before **`sudo`** is configured).

1.2.3. Formatting conventions

Commands intended for shell execution, as well as configuration file text, are featured in a **`bold monospace font`**. *`Italics`* are used to indicate instances where the system administrator must substitute the appropriate information into a command or configuration file. For example:

```
# ssh -l firstname.lastname somebox.mcgill.ca
Password:
```

```
[firstname.lastname@somebox ~]$ sudo su -
[sudo] password for firstname.lastname:
[root@somebox ~]# whoami
root
[root@somebox ~]#
```

There are also three visual styles to draw attention to information that might otherwise be overlooked.

Note

A note is a tip or shortcut or alternative approach to the task at hand. Ignoring a note should have no negative consequences, but you might miss out on a trick that makes your life easier.

Important

Important boxes detail things that are easily missed: configuration changes that only apply to the current session, or services that need restarting before an update will apply. Ignoring Important boxes won't cause data loss but may cause irritation and frustration.

Warning

A Warning should not be ignored. Ignoring warnings will most likely cause data loss, security holes, or will lock you out of a system.

1.2.4. Reboot required

A system reboot is implicitly required after some actions in order to complete the reconfiguration of the system. In many cases, the changes will not take effect until a reboot is performed. In order to ensure that changes are applied properly and to test functionality, always reboot the system after applying a set of recommendations from this guide.

If you are not sure what services will be started when a system is rebooted, then your system is not production-ready.

System-wide configuration

2.1. Installing and maintaining software

The following sections contain information on security-relevant choices during the initial operating system installation process and the setup of software updates.

2.1.1. Initial installation recommendations

The recommendations here apply to a clean installation of the system, where any previous installations are wiped out. The sections presented here are in the same order that the installer presents, but only installation choices with security implications are covered. Many of the configuration choices presented here can also be applied after the system is installed.

2.1.2. Disk Partitioning

It is important to create separate partitions that will be used by temporary files and logs. In addition to the partitions that you will need to create for the purposes of your installation, please create separate partitions for the following mount points:

```
/var/log
/var/tmp
/tmp
```

If your system is using VAS/QAS and will not be mounting a network home partition for users, you should also create a separate small **/home** partition to hold directories of users logging in.

Here is an example configuration for a web server with 40 GB total disk space (size in MB to match the installer):

Mount point	Size (MB)	Caveats
/boot	200	
swap	4096	
/	5000	more if you need java or similar large apps
/home	1000	unless **/home** is net-mounted
/var/log	5000	
/var/tmp	1000	
/tmp	1000	
/var/www	the rest	use "fill to maximum capacity" option

Table 2.1. Example partitioning for a webserver with a 40 GB disk

If you have more disk space available, or if you are expecting larger than usual amount of web traffic (and thus larger logs), you should increase the space allocated to the **/var/log** partition (15-20 GB should be enough even for high-traffic environments).

If you are using LVM2 on a large disk, a good habit is to leave 5-10 GB unallocated so in the future you can quickly grow a partition that is running out of space.

Note

By default, all created filesystems will have 5% of space reserved for superuser in order to prevent unprivileged processes from filling up the disk space. However, if you are creating very large partitions (especially over 1TB in size), you may want to adjust this setting using the **`tune2fs -m`** command. See **`man tune2fs`** for more information.

Important

If you store application-related data in **`/srv`** instead of the default locations (e.g. **`/var/www`** for httpd, or **`/var/lib/mysql`** for MySQL) you will need to make modifications to the daemon configurations, as well as add these locations to the SELinux labeling rules. It is recommended to leave application data in the default locations to minimize various risks.

Warning

Do not store application data in **`/home`**. It is bad practice and makes separating user data from application data and setting the correct MAC/RBAC labels needlessly difficult.

2.1.3. Bootloader configuration

Check the box to **Use a boot loader password** and create a password. Once this password is set, anyone who wishes to change the boot loader configuration will need to enter it. Assigning a boot loader password prevents a local user with physical access from altering the boot loader configuration at system startup.

2.1.4. Network devices

Unless use of DHCP is absolutely necessary, click the **Edit** button and:

1. Uncheck **Use Dynamic IP configuration (DHCP)**.
2. Uncheck **Enable IPv6 Support**
3. Enter appropriate IPv4 networking information as required.

With the DHCP setting disabled, the hostname, gateway, and DNS servers should then be assigned on the main screen.

2.1.5. Root password

The security of the entire system depends on the strength of the root password. The password should be at least 12 characters long, and should include a mix of capitalized and lowercase letters, special characters, and numbers. It should also not be based on any dictionary word.

2.1.6. Software packages

Uncheck all the package groups and install only the core system. This should be the case even if you are installing a "web server" — default configurations will pull in too many packages that you will not end up using, and that goes against the "do not install software you will not need" general guideline. You can install the precise packages you require after the installation is complete, by using "**`yum install`**."

2.1.7. Use kickstarts if possible

Red Hat provides a way to automate a lot of installation-related tasks via "kickstarts" — a specially formatted plaintext file loaded by the system prior to installation that act much like "installation recipes."

You can find out more about writing kickstart files in the *Red Hat Installation Guide*, available from the *Red Hat website*[1].

2.2. Modify the system login banners

The contents of the file **`/etc/issue`** are displayed on the screen just above the login prompt for users logging directly into a terminal. The contents of the file **`/etc/issue.net`** are displayed to anyone connecting to network services on that machine.

Any system at McGill should be displaying the following banner to anyone logging in to the system:

> **Disclaimer**
>
> All McGill resources are governed by the Policy on the Responsible use of McGill Information Technology Resources. This system is for the use of Authorized Users only.
>
> Individuals using McGill IT Resources in breach of McGill's policies and procedures or in excess of their authority are subject to having their activities monitored and recorded by System Administrators. In the course of monitoring individuals improperly using McGill IT Resources, or in the course of McGill IT Resources maintenance, the activities of Authorized Users could also be monitored. Anyone using this system expressly consents to such monitoring and is advised that if such monitoring reveals possible evidence of illegal activity, system personnel may provide the evidence of such monitoring to law enforcement officials.
>
> A violation of the provisions of this policy may constitute a disciplinary offence and, where appropriate, shall be dealt with under the regulations, policies, code or collective agreement to which the Authorized User is subject.

Please put this text in **`/etc/issue`** and **`/etc/issue.net`**.

[1] http://www.redhat.com/docs/manuals/enterprise/

2.3. System updates

All software has bugs — many of them lead to security vulnerabilities. It is extremely important to keep your systems patched to keep them immune to attacks.

Red Hat releases regular patches, usually falling into two main categories: security updates and bugfixes. We strongly advise promptly installing all security updates categorized as "Moderate" and "Important." Failure to apply security updates will result in system vulnerabilities that can and will eventually be exploited.

The fact that a system is behind a firewall or is located in private IP space does not mean that the system does not require patching. As stated previously, "defence in depth" is one of the main tenets of IT security. Errors in network and firewall design or maintenance can result in systems suddenly being available to external traffic, and we need to be prepared for such events — no matter how unlikely they may seem.

2.3.1. Do not install software "from sources"

Software installed "from sources" — compiled and installed on a system using "**`./configure`**", "**`make`**" and "**`make install`**" will quickly become difficult to maintain, especially if it has a large number of dependencies or is installed on a large number of systems. Most common software is already packaged and is available from Red Hat and should be installed from central package repositories provided by them. Packages provided by Red Hat are tested to make sure that they work well with other versions of libraries available on the system and receive timely security and bugfix updates.

> **Note**
>
> If some software you need is not available in stock RHEL5, you may find it available in *Fedora EPEL*, which stands for "Extra Packages for Enterprise Linux." Note that Fedora EPEL is only marginally affiliated with Red Hat and any packages you install from EPEL repositories will be packaged and maintained by volunteers.
>
> *Learn more about Fedora EPEL*[2]

2.3.2. Register with RHN Satellite Server

McGill provides a *RHN Satellite Server* that should be used to apply updates and keep track of installed Red Hat systems on campus. In order to be able to register your system with RHN Satellite Server, you will first need to obtain an *activation key* from NCS Enterprise by emailing *`enterprise.ncs@mcgill.ca`* .

Once you have the activation key, perform the following actions in order to enrol your system with McGill's RHN Satellite Server:

```
# rpm -ivh http://anik.ncs.mcgill.ca/pub/rhn-org-trusted-ssl-cert-1.0-1.noarch.rpm
# rhnreg_ks --activationkey [activation key] \
  --serverUrl=https://anik.ncs.mcgill.ca/XMLRPC \
  --sslCACert=/usr/share/rhn/RHN-ORG-TRUSTED-SSL-CERT
```

RHN Satellite Server offers a rich interface to all the machines you have registered with the service, showing such details as outstanding bugfixes, applicable security errata, system availability status, etc. It also allows administrators to remotely schedule system updates and perform some of the maintenance tasks on multiple systems at once.

2.3.3. Update your systems

Systems can be updated either via the RHN Satellite Server, or via command line using the **yum** utility. You can use the "*yum-security*" plugin to limit the package list to just the security-related updates. This should help minimize the amount of updates you have to install while remaining protected against security vulnerabilities.

```
# yum --security update-minimal
```

The "yum-security" plugin is installed by default.

Always test updates in pre-production or development systems before applying them in production environments. All updates go through rigorous QA at Red Hat, but that does not mean that they won't cause problems in your particular configuration.

Important

Remember that all kernel updates require a reboot in order to actually switch to the updated version of the kernel. Schedule system reboots well in advance, but do not forego them if there are important security vulnerabilities in the running kernel.

2.3.4. Red Hat Network Daemon (rhnsd)

Enrolled systems communicate with the RHN Satellite Server via a daemon called **rhnsd**. By default, each system will connect to the satellite server every 4 hours to report the list of installed software and to download any scheduled updates or execute commands specified by the administrator. This interval can be changed in **/etc/sysconfig/rhn/rhnsd** but cannot be made any shorter than the hardcoded minimum of 60 minutes.

This is a very simple daemon and should be left enabled on all machines enrolled with either McGill's RHN Satellite Server or Red Hat's RHN proper.

2.4. Do not disable SELinux

Security Enhanced Linux (SELinux) is an implementation of MAC and RBAC controls on Linux. While it's a well-established and trusted technology, a lot of administrators are reluctant to enable it because the default rules tend to conflict with third-party software, custom written legacy scripts, or they object to it purely because it alters the fundamental "Unix way" access permissions are handled in Linux.

Our recommendation is to leave SELinux enabled in *permissive mode*.

2.4.1. Enforcing vs. Permissive mode

SELinux has three basic states:

State	Description
enabled, enforcing	SELinux is enabled and is enforcing. The policies are loaded and anything that isn't allowed in the policy definition will be denied and logged. This mode is also called "IPS mode."
enabled, permissive	SELinux is enabled, but is not enforcing. The policies are loaded, but any policy violations will be only logged and then allowed. In this mode, SELinux does not interfere with legacy "Unix-way" operation of Linux. This mode is also called "IDS mode."
disabled	SELinux is completely disabled.

Table 2.2. SELinux states

Additionally, RHEL5 ships with the following pre-written SELinux policies:

Policy	Description
selinux-policy-minimum	A minimum set of rules for writing your own policy. Not recommended for anyone without very advanced knowledge of SELinux.
selinux-policy-strict	A rigid set of rules for a very tight lock-down of system processes. Anything not explicitly allowed will be denied. Not recommended for any environment lacking advanced SELinux expertise.
selinux-policy-mls	Implements NSA's very rigid "Multi-Layer Security" policy. MLS is a great theoretical approach to handling classified information, but has significant practical drawbacks. Not recommended unless your system must for some reason meet NSA guidelines.
selinux-policy-targeted	Most commonly used SELinux policy that is installed by default. The goal of this policy is to label and lock down most common system daemons and files. All user processes are executed in the "unconfined" space that relies only on the "legacy" permissions framework (read-write-execute permissions for users and groups).

Table 2.3. SELinux default policies

We recommend leaving SELinux enabled in *permissive mode* with a *targeted policy*. While this doesn't lend any additional security offered by SELinux to the system, it allows for full auditing of unusual process activity and can be used for both intrusion detection and forensic purposes.

To configure SELinux in enabled-permissive mode, modify **`/etc/sysconfig/selinux`** as follows:

```
SELINUX=permissive
SELINUXTYPE=targeted
```

Alternatively, you can run **`system-config-securitylevel-tui`** command to achieve the same goal.

> **Important**
>
> If you decide to enable SELinux on a system where it was previously completely disabled, you will need to relabel the entire filesystem upon system reboot. This can take a *very long time* on filesystems that contain a lot of files or that are accessed via NAS. It is therefore not recommended to change SELinux status of systems already in production.
>
> If you do decide to enable SELinux on a system where it has been previously disabled, it is best to use **`system-config-securitylevel-tui`** command because it will mark all the necessary filesystems for relabeling. A system will require a reboot to change SELinux status and relabel files.

2.4.2. SELinux-related services

There are a number of SELinux-related daemons installed and started by default. Following is a recommendation on which ones to leave enabled and which ones to turn off.

2.4.2.1. SETroubleshoot

This service is only installed with a GUI desktop environment. It helps administrators troubleshoot SELinux permission problems, but it is unnecessary and even meddling on anything that isn't a system administrator's desktop. If you find that **`setroubleshoot`** is installed, you can turn it off using:

```
# chkconfig setroubleshoot off
```

You can additionally erase it entirely.

```
# yum erase setroubleshoot
```

2.4.2.2. MCS Translation service (`mcstrans`)

This service is only useful if you have installed or modified a policy that uses category labels, such as **selinux-policy-mls**. It is not needed in any other case and can be safely turned off.

```
# chkconfig mcstrans off
```

2.4.2.3. Restorecon service (`restorecond`)

The restorecond daemon monitors a list of files which are frequently created or modified on running systems, and whose SELinux contexts are not set correctly. It looks for creation events related to files listed in **`/etc/selinux/restorecond.conf`**, and sets the contexts of those files when they are discovered.

It's a useful service and should be left enabled.

2.4.3. Further information about SELinux

SELinux is a very broad topic that can't possibly be covered in a "setup guidelines" document. If you are interested in finding out more about this technology, you can consult the following resources:

- *NSA's SELinux page*[3]
- *Fedora Project's SELinux FAQ*[4]
- Red Hat Enterprise Linux Deployment Guide from Red Hat

Network configuration and firewalls

Most machines must be connected to a network of some sort, and this brings with it the substantial risk of network attack. This section discusses the security impact of decisions about networking which must be made when configuring a system.

3.1. Kernel tweaks for networking

The **sysctl** utility is used to set a number of parameters which affect the operation of the Linux kernel. Several of these parameters are specific to networking, and the configuration options in this section are recommended.

3.1.1. Network parameters for hosts only

Most systems do not need to forward any packets between networks, or act as gateways. Unless you are specifically configuring a router, make sure the following is set in **`/etc/sysctl.conf`**:

```
net.ipv4.ip_forward = 0
net.ipv4.conf.all.send_redirects = 0
net.ipv4.conf.default.send_redirects = 0
```

These settings will prevent hosts from executing any network functionality that is only appropriate for routers.

3.1.2. Network parameters for all systems

The following **`/etc/sysctl.conf`** settings should be configured on all systems:

```
net.ipv4.conf.all.accept_source_route = 0
net.ipv4.conf.all.accept_redirects = 0
net.ipv4.conf.all.secure_redirects = 0
net.ipv4.conf.all.log_martians = 1
net.ipv4.conf.default.accept_source_route = 0
net.ipv4.conf.default.accept_redirects = 0
net.ipv4.conf.default.secure_redirects = 0
net.ipv4.icmp_echo_ignore_broadcasts = 1
net.ipv4.icmp_ignore_bogus_error_responses = 1
net.ipv4.tcp_syncookies = 1
net.ipv4.conf.all.rp_filter = 1
net.ipv4.conf.default.rp_filter = 1
```

These options improve Linux’s ability to defend against certain types of IPv4 protocol attacks.

The `accept_source_route`, `accept_redirects`, and `secure_redirects` options are turned off to disable IPv4 protocol features which are considered to have few legitimate uses and to be easy to abuse.

The `log_martians` option logs several types of suspicious packets, such as spoofed packets, source-routed packets, and redirects.

The `icmp_echo_ignore_broadcasts` and `icmp_ignore_bogus_error_responses` options protect against ICMP attacks.

The `tcp_syncookies` option uses a cryptographic feature called SYN cookies to allow machines to continue to accept legitimate connections when faced with a SYN flood attack.

The `rp_filter` option enables RFC-recommended source validation. It should not be used on machines which are routers for very complicated networks, but is helpful for end hosts and routers serving small networks.

For more information on any of these, see the kernel source documentation.

3.2. IPv6

McGill is slowly moving towards enabling IPv6 on campus, but at the moment there are no enterprise-level services requiring IPv6. Unless it is absolutely necessary for you to be running IPv6 services, please disable IPv6 for all network interfaces.

3.2.1. Disable interface usage of IPv6

To prevent configuration of IPv6 for all interfaces, add or correct the following lines in **`/etc/sysconfig/network`**:

```
NETWORKING_IPV6=no
IPV6INIT=no
```

For each network interface *`IFACE`*, add or correct the following lines in **`/etc/sysconfig/network-scripts/ifcfg-`*`IFACE`*** as an additional prevention mechanism:

```
IPV6INIT=no
```

If it becomes necessary later to configure IPv6, only the interfaces requiring it should be enabled.

3.3. IPTables

A host-based firewall called Netfilter is included as part of the Linux kernel distributed with the system. It is activated by default. This firewall is controlled by the program iptables, and the entire capability is frequently referred to by this name. An analogous program called ip6tables handles filtering for IPv6. Netfilter filtering occurs at the kernel level, before a program can even process the data from the network packet. As such, any program on the system is affected by the rules written.

3.3.1. Default IPTables template for McGill

McGill University InfoSec provides a default iptables template that you can use on your systems. Please email *`infosec@mcgill.ca`* for the latest version of the file.

3.3.2. Editing and enabling IPTables rules

Place the template into **`/etc/sysconfig/iptables`** and edit according to comments in the template. If you require help configuring firewalls for your server, please email *`infosec@mcgill.ca`* for a member of Information Security team to assist you.

Once the iptables have been enabled, start the iptables service:

```
# /sbin/service iptables restart
# /sbin/chkconfig iptables on
```

This will ensure that the iptables are started and loaded after each system reboot.

Important

Part of the "iptables restart" script functionality is to unload and load the relevant iptables modules. However, as long as `ESTABLISHED,RELATED` rule is used, this will result in interruption of some of the established connections as the state-tracking iptables module is reloaded. This may have negative effects on production systems. If you are making small changes on a production system that do not add or remove any additional iptables modules (that is, you did not add any new, previously unused **`-m`** options), you should use the following command instead:

```
# /sbin/iptables-restore < /etc/sysconfig/iptables
```

3.3.3. Further strengthening

Further strengthening, particularly as a result of customization to a particular environment, is possible for the iptables rules. Consider the following options, though their practicality depends on the network environment and usage scenario:

- *Restrict outgoing traffic.* The OUTPUT chain’s default policy can be changed to DROP, and rules can be written to specifically allow only certain types of outbound traffic. Such a policy could prevent casual usage of insecure protocols such as ftp and telnet, or even disrupt spyware. However, it would still not prevent a sophisticated user or program from using a proxy to circumvent the intended effects, and many client programs even try to automatically tunnel through port 80 to avoid such restrictions.

- *SYN flood protection.* SYN flood protection can be provided by iptables, but might run into limiting issues for servers. For example, the `iplimit` match can be used to limit simultaneous connections from a given host or class. Similarly, the `recent` match allows the firewall to deny additional connections from any host within a given period of time (e.g. more than 3 `--state NEW` connections on port 22 within a minute to prevent dictionary login attacks). Note, though that most commonly we see distributed attacks that are specifically crafted to avoid triggering the limit rules.

- *Port-knocking.* The default McGill template does not allow any hosts from outside of McGill networks to connect to the ssh port. At times, it is useful to be able to ssh in from external hosts in cases when there are no other ways for an administrator to gain access and VPN option is not available. Port-knocking, if configured, would allow a remote connection to the ssh port if at first a connection attempt is made to another pre-established closed port. The following rule will open port 22 to the computer that first attempts to connect to high port *`NNNN`* (no default port number is set here as this is supposed to be only known to administrators of that particular machine):

  ```
  # Allow port-knocking on port NNNN to open ssh
  ```

```
-A NET-GLOBAL-INPUT -m state --state NEW -m tcp -p tcp --dport 22 -m recent --rcheck --name ssh
 -j ACCEPT
-A NET-GLOBAL-INPUT -m state --state NEW -m tcp -p tcp --dport NNNN -m recent --name ssh --set
 -j LOG-AND-DROP
# The ports around NNNN re-close 22
-A NET-GLOBAL-INPUT -m state --state NEW -m tcp -p tcp -m multiport --dports NNNM,NNNO -m
 recent --name ssh --remove -j LOG-AND-DROP
```

See more information about port-knocking in this *Wikipedia article*[1].

3.4. TCP Wrapper

TCP Wrapper is a library which provides simple access control and standardized logging for supported applications which accept connections over a network. TCP Wrapper supports only services which were built to make use of the libwrap library. To determine whether a given executable daemon **`/path/to/daemon`** supports TCP Wrapper, check the documentation, or run:

```
# ldd /path/to/daemon | grep libwrap.so
```

If this command returns any output, then the daemon probably supports TCP Wrapper.

> **Note**
>
> If you are already using IPTables in addition to network-level router ACLs, the additional layer of TCP wrappers may not provide enough benefit for it to be worth your while. If, on the other hand, your configuration does not rely on any router ACLs, you may want to use TCP wrappers as added protection for the cases when your IPTables firewalls are for some reason disabled.

3.4.1. How TCP Wrapper protects services

TCP Wrapper provides access control for the system's network services using two configuration files. When a connection is attempted:

- The file **`/etc/hosts.allow`** is searched for a rule matching the connection. If one is found, the connection is allowed.
- Otherwise, the file **`/etc/hosts.deny`** is searched for a rule matching the connection. If one is found, the connection is rejected.
- If no matching rules are found in either file, then the connection is allowed. By default, TCP Wrapper does not block access to any services.

In the simplest case, each rule in **`/etc/hosts.allow`** and **`/etc/hosts.deny`** takes the form:

```
daemon: client
```

where `daemon` is the name of the server process for which the connection is destined, and `client` is the partial or full hostname or IP address of the client. It is valid for daemon and client to contain one item, a comma-separated list of items, or a special keyword like ALL, which matches any service or client. (See the **hosts.access(5)** manpage for a list of other keywords.)

Partial hostnames start at the root domain and are delimited by the **.** character. So the client machine *example1.dept.mcgill.ca*, with IP address *10.7.2.3*, could be matched by any of the specifications:

```
.mcgill.ca
.dept.mcgill.ca
10.7.2.
```

3.4.2. Restrict all connections by default

Restrict all connections to non-public services to localhost only. For example, for a system that has **vsftpd** accepting incoming connections from everywhere and **sshd** from your department only, you may want to put the following in **/etc/hosts.allow**:

```
vsftpd: ALL
sshd: .dept.mcgill.ca
ALL: localhost
```

And in **/etc/hosts.deny**:

```
ALL: ALL
```

Warning

Note that this is an example only. You will also need to provide network IP ranges for your department in addition to using domain wildcards, unless all hosts in your domain have valid reverse DNS mapping.

Chapter 4.

Logging and auditing

Successful local or network attacks on systems do not necessarily leave clear evidence of what happened. It is necessary to build a configuration in advance that collects this evidence, both in order to determine that something anomalous has occurred, and in order to respond appropriately. In addition, a well-configured logging and audit infrastructure will show evidence of any misconfiguration which might leave the system vulnerable to attack.

Logging and auditing take different approaches to collecting data. A logging infrastructure provides a framework for individual programs running on the system to report whatever events are considered interesting: the **`sshd`** program may report each successful or failed login attempt, while the **`sendmail`** program may report each time it sends an e-mail on behalf of a local or remote user. An auditing infrastructure, on the other hand, reports each instance of certain low-level events, such as entry to the setuid system call, regardless of which program caused the event to occur.

Auditing has the advantage of being more comprehensive, but the disadvantage of reporting a large amount of information, most of which is uninteresting. Logging (particularly using a standard framework like syslogd) has the advantage of being compatible with a wide variety of client applications, and of reporting only information considered important by each application, but the disadvantage that the information reported is not consistent between applications.

A robust infrastructure will perform both logging and auditing, and will use configurable automated methods of summarizing the reported data, so that system administrators can remove or compress reports of events known to be uninteresting in favor of alert monitoring for events known to be interesting.

This section discusses how to configure logging, log monitoring, and auditing, using tools included with RHEL5. We will be using **`syslog`** for logging and **`auditd`** for auditing.

4.1. Configure syslog for remote logging

McGill provides a centralized log capturing server that should be used on all enterprise machines. To enable, please append the following to your **`/etc/syslog.conf`**:

```
# remote logging to the log server
*.*                       @slurp.ncs.mcgill.ca
```

Now we need to make sure that the logging continues even if DNS becomes unavailable. For that purpose, add the following to the **`/etc/hosts`** file:

```
132.216.30.204          slurp.ncs.mcgill.ca
```

The central logserver processes the logs for monitoring purposes, so unless you want notifications of your server activity in addition to what is already monitored by slurp, you may want to remove **`logwatch`** from the server by running:

```
# yum -y remove logwatch
```

4.2. System accounting with auditd

The audit service is provided for system auditing. By default, the service audits about SELinux AVC denials and certain types of security-relevant events such as system logins, account modifications, and authentication events performed by programs such as sudo.

Under its default configuration, auditd has modest disk space requirements, and should not noticeably impact system performance. The audit service, configured with at least its default rules, is strongly recommended for all sites, regardless of whether they are running SELinux.

4.2.1. Enable the `auditd` service

Ensure that the auditd service is enabled (this is the default):

```
# chkconfig auditd on
```

By default, **`auditd`** logs only SELinux denials, which are helpful for debugging SELinux and discovering intrusion attempts, and certain types of security events, such as modifications to user accounts (**`useradd`**, **`passwd`**, etc), login events, and calls to **`sudo`**.

Data is stored in **`/var/log/audit/audit.log`**. By default, **`auditd`** rotates 4 logs by size (5MB), retaining a maximum of 20MB of data in total, and refuses to write entries when the disk is too full. This minimizes the risk of audit data filling its partition and impacting other services.

4.2.2. Configure `auditd` data retention

The total amount of space used by **`auditd`** will be `max_log_file` times `num_logs`, so our configuration should reflect that. With the auditing rules that we are going to be using, a relatively busy system will generate 3-10 MB of logs daily, so to keep roughly a month's worth of logs we recommend changing the following setting in **`/etc/audit/auditd.conf`**:

```
max_log_file = 50
num_logs = 5
```

This should fill up at most 250 MB of space on your **`/var/log`** partition. This should be sufficient for most configurations, but these settings can be tweaked further if the audit logs are filling up too quickly.

4.2.3. Configure `auditd` rules

Note

McGill University InfoSec provides a default template for **`audit.rules`** that you can use on your systems. It is based on the STIG ruleset. Please email *`infosec@mcgill.ca`* for the latest version of the file.

The auditd program can perform comprehensive monitoring of system activity. We will be using a set of rules as recommended by the NSA (STIG). Apply them by copying the bundled STIG rules into place:

```
# cp /usr/share/doc/audit-*/stig.rules /etc/audit/audit.rules
```

Important

If you are running an i386 server, comment out all lines that mention `arch=b64`. You do not need to comment out `arch=b32` on 64-bit systems, as they provide 32-bit syscalls as well.

The default configuration is fairly robust, though you may want to change the following two settings (at the beginning and at the end of **`audit.rules`**:

- **`-f`**: The STIG rules set failure mode to "panic", meaning that if the audit daemon fails, the machine will shut down. This is the most secure configuration that makes sense for fine-tuned production systems requiring full protection against audit tampering, but it may not be what you want when you are just installing the system, or when the system does not require such strict controls. Change the **`-f`** parameter to **`1`** to go down to "console alert" mode.

- **`-e`**: The STIG rules set it to "2" which means that the rules are set as "immutable" — once the rules are read, the machine will require a restart in order to modify them. This setting makes sense for fine-tuned production systems, but while you are still configuring or tuning the audit rules, you should change the **`-e`** setting to **`1`**.

You can also use the audit daemon to track changes to various files, such as configuration files. For example, if you want to log whenever a change is made to any files in **`/etc/httpd`**, you can add the following to the audit rules:

```
# Track modified files in /etc/httpd
-w /etc/httpd -p aw -k config-httpd
```

The **`-k config-httpd`** option tags all audit logs that matched this configuration with a "config-httpd" key so you can easily review the changes using **`aureport`**.

4.2.4. Send audit logs to syslog

The auditing daemon will only log to the local file by default. If your system is configured to send logs to McGill's central log aggregator (and it should be), then you will need to configure **`auditd`** to send all events to syslog as well. Edit **`/etc/audisp/plugins.d/syslog.conf`** and change "**`active = no`**" to "**`active = yes`**".

4.2.5. Restart the `auditd` daemon

Remember to restart the **`auditd`** daemon whenever you modify the rules. Unless you have specified **`-e 2`** in **`audit.rules`**, you should be able to do it by running:

```
# /sbin/service auditd restart
```

If you did specify "**`-e 2`**" (or forgot to change it to 1), you will need to restart the system in order for the changes to take effect.

4.2.6. Reviewing audit information with aureport

The audit logs provide a lot of information, and to make it useful you will need to familiarize yourself with **aureport** and **ausearch** commands. The manpages for both are fairly extensive, but here is a couple of quick commands to get you started.

To show a quick count of all logged messages sorted by the "key" specified in the configuration (audit.rules):

```
# aureport --key --summary
```

Show all "access denied" hits:

```
# ausearch --key access --raw | aureport --file --summary
```

If you have added the command to log all changes to config files ("config-httpd" in our example), this will produce a "file report" with date, time, name of the file, type of access, executable, and the original username of the user making the change ("original" or "audit username" means that if someone used **su** or **sudo** to switch to root or some other user, the log will still show the username they logged in with):

```
# ausearch --key config-httpd --raw | aureport --file -i
```

You can use **-ts** and **-te** flags to **aureport** and **ausearch** that will let you limit your searches to a specific time frame, for example:

```
# ausearch --key config-httpd -ts yesterday --te today --raw | aureport --file -i
```

When SELinux is enabled, it also uses **auditd** for logging, in addition to using **syslogd**.

4.2.7. Using pam_tty_audit to log all keystrokes

If you want to log all keystrokes performed by a user logging in or switching to a different account, you could use **pam_tty_audit**. You can read more information about this auditing tool using "**man pam_tty_audit**." To enable it on a system, you will need to modify a number of files in **/etc/pam.d**.

First, create **/etc/pam.d/pam_tty_audit** with the following contents:

```
#%PAM-1.0
# This enables keystroke auditing for root user
session required pam_tty_audit.so disable=* enable=root open_only
```

Next, append the following line to **/etc/pam.d/sudo** and **/etc/pam.d/sudo-i**:

```
session include pam_tty_audit
```

You can review keystroke data by using the following command:

```
# aureport --tty -i
```

Warning

This is very intrusive auditing and will require prior approval from CIO or Legal Services. You may be required to place a warning in **`/etc/issue.net`** to advise administrators that all their keystrokes are being logged. Additionally, such blanket collection of keystroke data may result in someone's password being stored in a logfile or even transmitted via the network, if the audit daemon is configured to send log data to an external syslog server. Please do not enable keystroke logging without considering all implications of such action.

4.3. Audit access with sudo

The **sudo** command allows fine-grained control over which users can execute commands using other accounts. The primary benefit of **sudo** over **su** or logging in as user root is that it provides an audit trail of every command run by a privileged user. It is possible for a malicious administrator to circumvent this restriction, but, if there is an established procedure that all root commands are run using sudo, then it is easy for an auditor to detect unusual behavior when this procedure is not followed.

Editing **`/etc/sudoers`** by hand can be dangerous, since a configuration error may make it impossible to access the root account remotely. The recommended means of editing this file is using the **`visudo`** command, which checks the file’s syntax for correctness before allowing it to be saved.

Note

On a system with **Quest QAS** (formerly **Vintela VAS**) installed, the **`sudoers`** file is located in **`/etc/opt/quest/sudo/sudoers`** and **`/etc/sudoers`** is ignored. The **quest-sudo** package provides their own **`visudo`** binary that should look for it in the correct VAS-specific location.

As a basic precaution, *never* use the **`NOPASSWD`** directive, which would allow anyone with access to an administrator account to execute commands as root without knowing the administrator’s password.

The sudo command has many options which can be used to further customize its behavior. See the **`sudoers(5)`** man page for details.

Services

5.1. Disable All Unneeded Services at Boot Time

The best protection against vulnerable software is running less software. This section describes how to review the software which Red Hat Enterprise Linux installs on a system and disable software which is not needed. It then enumerates the software packages installed on a default RHEL5 system and provides guidance about which ones can be safely disabled.

5.1.1. Determine which Services are Enabled at Boot

Run the command:

```
# chkconfig --list | grep :on
```

The first column of this output is the name of a service which is currently enabled at boot. Review each listed service to determine whether it can be disabled.

If it is appropriate to disable some service srvname , do so using the command:

```
# chkconfig srvname off
```

In addition to turning a service off you may also want to completely remove that service from the system. To determine which RPM package owns the service in question, you may run the following command:

```
# rpm -qf /etc/init.d/srvname
```

Once you determine the name of the package, you may remove it from the system using **yum**:

```
# yum remove pkgname
```

Warning

Carefully review the list of dependencies before removing any packages. Some services are provided by core packages that may not be removed without rendering the system inoperable. Review *Section A.2, “Services to disable checklist”* for guidance on which packages are safe to remove.

Use the guidance below for information about unfamiliar services.

5.1.2. Guidance on Default Services

The table in this section contains a list of all services which should be enabled at boot in a default RHEL5 installation. For each service, one of the following recommendations is made:

- *Enable:* The service provides a significant capability with limited risk exposure. Leave the service enabled.
- *Configure:* The service either is required for most systems to function properly or provides an important security function. It should be left enabled by most environments. However, it must be configured securely on all machines, and different options may be needed for workstations than for servers. See the referenced section for recommended configuration of this service.

All other running services should be disabled, unless the system is specifically built as a dedicated server to run that service (e.g. if you are configuring a DNS server and must thus run **named**). In this case, please familiarize yourself with that software's "best practices" information, or refer to other "system hardening" documentation such as the NSA hardening guide.

Service name	Action	Reference
atd	Configure	*Section 5.3, "**Cron** and **At** daemons"*
auditd	Configure	*Section 4.2, "System accounting with auditd"*
crond	Configure	*Section 5.3, "**Cron** and **At** daemons"*
iptables	Configure	*Section 3.3, "IPTables"*
irqbalance	Enable on SMP systems	*Section 5.2.1, "Interrupt Distribution on Multiprocessor Systems (**irqbalance**)"*
lvm2-monitor	Disable, unless needed	*Section 5.2.2, "Monitor for mirrored LVM partitions (**lvm2-monitor**)"*
messagebus	Enable	*Section 5.2.3, "D-Bus IPC Service (**messagebus**)"*
microcode_ctl	Enable on IA32 systems	*Section 5.2.4, "IA32 Microcode Utility (**microcode_ctl**)"*
network	Configure	*Chapter 3, Network configuration and firewalls*
ntpd	Configure	*Section 5.4, "Network Time Protocol Daemon (**ntpd**)"*
restorecond	Enable	*Section 2.4.2.3, "Restorecon service (**restorecond**)"*
rhnsd	Enable	*Section 2.3.4, "Red Hat Network Daemon (**rhnsd**)"*
postfix	Configure	*Section 5.5, "Mail Transfer Agent"*
psacct	Enable	*Section 5.6, "Process accounting daemon (**psacct**)"*
smartd	Disable on VM guests	*Section 5.2.5, "SMART disk monitoring (**smartd**)"*
syslog	Configure	*Section 4.1, "Configure syslog for remote logging"*
xinetd	Enable if NetBackup is used	*Section 5.2.6, "Extended Internet Services Daemon (**xinetd**)"*
McGill Infrastructure Services		
spong-client	Enable	Configured by NCS

McGill Infrastructure Services		
sshd-quest	Configure	Configured by NCS
vasd	Configure	Configured by NCS

Table 5.1. Default enabled services

All other running services should be disabled, unless absolutely needed (refer to the *Section A.2, “Services to disable checklist”*).

5.2. Base services

5.2.1. Interrupt Distribution on Multiprocessor Systems (`irqbalance`)

The goal of the irqbalance service is to optimize the balance between power savings and performance through distribution of hardware interrupts across multiple processors.

In a server environment with multiple processors, this provides a useful service and should be left enabled. If a machine has only one processor, the service may be disabled:

```
# chkconfig irqbalance off
```

5.2.2. Monitor for mirrored LVM partitions (`lvm2-monitor`)

LVM provides a mirroring functionality that allows multiple physical volumes to be used as mirrors for the same logical volume. The monitoring daemon, **`lvm2-monitor`** exists to switch from one physical volume to another in the event of a failure.

Unless you are using lvm2 mirroring (and you are probably not, as most enterprise systems rely on hardware raid for similar failover purposes), disable lvm2-monitor:

```
# chkconfig lvm2-monitor off
```

5.2.3. D-Bus IPC Service (`messagebus`)

More and more services require D-Bus to communicate between processes. Unless you are sure that none of the services running on your system require D-Bus, leave the **`messagebus`** service enabled. You can look in **`/etc/dbus-1/system.d`** to see what software installed on your system relies on D-Bus.

5.2.4. IA32 Microcode Utility (`microcode_ctl`)

`microcode_ctl` is a microcode utility for use with Intel IA32 processors (Pentium Pro, PII, Celeron, PIII, Xeon, Pentium 4, etc). If the system is not running an Intel IA32 processor, disable this service:

```
# chkconfig microcode_ctl off
```

5.2.5. SMART disk monitoring (`smartd`)

SMART (Self-Monitoring, Analysis, and Reporting Technology) is a feature of hard drives that allows them to detect symptoms of disk failure and relay an appropriate warning. This technology is considered to bring relatively low security risk, and can be useful on non-virtual systems.

This daemon is useless on VM guests and should be turned off.

5.2.6. Extended Internet Services Daemon (`xinetd`)

Xinetd is unused by the default installation of RHEL5, but the NetBackup software used at McGill uses it for remote backups. If your system is going to have NetBackup installed, then you will need to leave xinetd enabled.

5.3. `Cron` and `At` daemons

The **`cron`** and **`at`** services are used to allow commands to be executed at a later time. The **`cron`** service is required by almost all systems to perform necessary maintenance tasks, while **`at`** may or may not be required on a given system. Both daemons should be configured defensively.

5.3.1. Disable `atd` if possible

The **`at`** command allows scheduling of other commands to be run at a specified date or time. As opposed to **`cron`**, it doesn't provide a facility for recurring actions and thus is rarely required on most systems unless the administrator must have the ability to execute scheduled commands at specific times.

Unless this functionality is required, disable the **`atd`** daemon by executing:

```
# chkconfig atd off
```

5.3.2. Restrict `at` and `cron` to Authorized Users

- Remove the **`/etc/cron.deny`** file.
- Edit **`/etc/cron.allow`**, adding one line for each user allowed to use the **`crontab`** command to create cron jobs.
- Remove the **`/etc/at.deny`** file.
- Edit **`/etc/at.allow`**, adding one line for each user allowed to use the **`at`** command.

The **`/etc/cron.allow`** and **`/etc/at.allow`** files contain lists of users who are allowed to use cron and at to delay execution of processes. If these files exist and if the corresponding files **`/etc/cron.deny`** and **`/etc/at.deny`** do not exist, then only users listed in the relevant allow files can run the **`crontab`** and **`at`** commands to submit jobs to be run at scheduled intervals.

On many systems, only the system administrator needs the ability to schedule jobs. Note that even if a given user is not listed in **`cron.allow`**, cron jobs can still be run as that user. The **`cron.allow`** file controls only administrative access to the **`crontab`** command for scheduling and modifying cron jobs.

5.3.3. Use `/etc/cron.d`

Instead of using **`crontab`** to set up cron jobs, consider placing configuration files into **`/etc/cron.d`** directory instead. This will allow you to duplicate configuration files among multiple machines when necessary, and back them up correctly with other configuration files. The format of the cron files is very similar to crontab, except there is an extra field between the time specification and the command, where you need to provide the user with whose privileges the command should be executed.

For examples, see **`man 5 crontab`**, the *Jobs in /etc/cron.d* section.

Please also keep in mind that RHEL 5 provides the following directories to simplify routine maintenance jobs:

```
/etc/cron.hourly
/etc/cron.daily
/etc/cron.weekly
/etc/cron.monthly
```

You can just place a script into these directories to have it executed with the frequency indicated (scripts will execute with root privileges).

5.4. Network Time Protocol Daemon (`ntpd`)

The Network Time Protocol is used to manage the system clock over a network. Computer clocks are not very accurate, so time will drift on unmanaged systems. Central time protocols can be used both to ensure that time is consistent among a network of machines, and that their time is consistent with the outside world.

Local time synchronization is recommended for all networks. If every machine on your network reliably reports the same time as every other machine, then it is much easier to correlate log messages in case of an attack. In addition, a number of cryptographic protocols (such as Kerberos, and, by extension, VAS/QAS) use timestamps to prevent certain types of attacks. If your network does not have synchronized time, these protocols may be unreliable or even unusable.

McGill provides local ntp servers you can use. To configure your ntp client to use McGill's servers, change the **`/etc/ntp.conf`** configuration file to use the following two servers:

```
server splish.cc.mcgill.ca
server splash.cc.mcgill.ca
```

Note

VAS/QAS can also synchronize time with the Active Directory, but will correctly recognize if ntpd is already configured on the system and will turn off that functionality.

5.5. Mail Transfer Agent

Every server requires a Mail Transfer Agent (MTA) for the purposes of either sending or receiving email. McGill provides a central MTA, so this guide only covers the configuration of a local relay that will deliver messages originating on your server to the intended recipients.

5.5.1. Install Postfix

Though RHEL 5 comes with Sendmail as the default MTA software, we recommend that you use Postfix instead. Sendmail has a history of security flaws, and Postfix was written with security in mind as a replacement for Sendmail. Though both services can be configured securely, certain features of Postfix make it a more appealing package.

To install Postfix, run the following commands:

```
# yum install postfix
# yum erase sendmail
```

5.5.2. Configure postfix

The default postfix configuration will only accept connections via localhost, which is sufficient for the purposes of relaying local mail. Use the following command to specify that postfix should use McGill's main mail routing service to relay the email:

```
# postconf -e relayhost=mailhost.mcgill.ca
```

Now configure postfix to start at boot:

```
# chkconfig postfix on
```

5.5.3. Configure root alias

Edit **`/etc/aliases`** to make sure that all email sent to root user on that machine is delivered to the system administrator email. At the end of the file, uncomment the root alias and set the destination email:

```
root: email@mcgill.ca
```

You will need to run **`newaliases`** command whenever you modify the aliases file:

```
# newaliases
```

You can test email delivery by using the **`mail`** command:

```
# echo test | mail -s test root
```

You can look in **`/var/log/maillog`** to see whether the delivery was successful.

5.6. Process accounting daemon (psacct)

Process accounting daemon logs data associated with processes running on the system, including commands issued by users. It is installed on all RHEL5 systems by default, but is not enabled. You can enable it by running:

```
# chkconfig psacct on
```

To view the list of commands recently executed by users and processes use **lastcomm**. Other commands, such as **sa** and **ac** give additional summary information about processes and connection times. See manpages for all of the above commands for more information.

5.7. Configuring OpenSSH (sshd)

The SSH protocol is recommended for remote login and remote file transfer. SSH provides confidentiality and integrity for data exchanged between two systems, as well as server authentication, through the use of public key cryptography. The implementation included with the system is provided by OpenSSH.

> **Note**
>
> Note that the same is true for a VAS/QAS-enabled system. The package **quest-openssh** adds a few patches to make it work with VAS/QAS but it's pretty much just standard OpenSSH installed in a different location.
>
> The configuration files for **quest-openssh** are in **/etc/opt/quest/ssh** instead of the default **/etc/ssh**. The configuration parameters mentioned below apply to both the stock **openssh** and the modified **quest-openssh**, but **quest-openssh** will ignore the configuration parameters in **/etc/ssh**, so make sure you are editing the right file.

5.7.1. Disable root logins

The default configuration allows logging in with root user credentials. Once **sudo** is configured, this should be disabled to make sure that all administrative actions are appropriately logged for auditing purposes.

To disable root login via SSH, add or correct the following line in **/etc/ssh/sshd_config**:

```
PermitRootLogin no
```

5.7.2. Limit users' SSH access

You should use **AllowUsers** or **AllowGroups** directives to limit who is able to ssh to your system. Modify **/etc/ssh/sshd_config** to reflect your needs accordingly.

Note

On VAS/QAS-enabled systems this is better accomplished via **vasd**'s configuration parameters in **`/etc/opt/quest/vas/users.allow`**.

5.7.3. Display login banner

To enable the display of the login banner (see *Section 2.2, “Modify the system login banners”*), you will need to enable the **`Banner`** option:

```
Banner /etc/issue.net
```

5.8. McGill infrastructure services

Systems that are centrally supported by McGill Enterprise Systems will additionally have the following software installed NCS:

Software	Description
Spong	Spong is a network monitoring system written in Perl. It requires that a local client is installed in order to collect information about the running system and to report usage patterns. It must be installed by NCS.
VAS/QAS	**Quest QAS** (formerly **Vintela VAS**) is a commercial package that allows joining Linux system to Active Directory. It mostly relies on Open-Source components such as winbind, kerberos and pam in order to achieve this. VAS/QAS requires licenses in order to operate and must thus be installed by NCS.
RSA SecurID	RSA SecurID is a commercial two-factor authentication implementation from RSA Security. It requires that someone has a physical device (usually a keyfob token) issued by McGill in order to be able to successfully authenticate to a system. RSA SecurID must be installed and configured by NCS and InfoSec.
NetBackup	McGill uses Veritas NetBackup for its centralized backup system. It requires licensing and must be installed by NCS Enterprise group.

Table 5.2. Software installed by NCS Enterprise

Configuration of these services is beyond the scope of this guide.

Appendix A. Checklists

A.1. Install and post-install checklists

	Description	Reference
□	Separate `/var/log`, `/var/tmp`, `/tmp`	*Section 2.1.2, “Disk Partitioning”*
□	Bootloader is password-protected	*Section 2.1.3, “Bootloader configuration”*
□	DHCP and IPv6 are not used	*Section 2.1.4, “Network devices”*
□	Strong root password is used	*Section 2.1.5, “Root password”*
□	Minimal software is installed	*Section 2.1.6, “Software packages”*

Table A.1. Installation checklist

	Description	Reference
□	System is registered with RHN	*Section 2.3.2, “Register with RHN Satellite Server”*
□	Latest security updates are installed	*Section 2.3.3, “Update your systems”*
□	**SELinux** is enabled in permissive mode	*Section 2.4, “Do not disable SELinux”*
□	Kernel networking parameters are set	*Section 3.1, “Kernel tweaks for networking”*
□	IPv6 is disabled	*Section 3.2, “IPv6”*
□	Login banners have been modified	*Section 2.2, “Modify the system login banners”*
□	Infosec's **`iptables`** template is installed	*Section 3.3, “IPTables”*
□	**Syslog** is configured to log remotely	*Section 4.1, “Configure syslog for remote logging”*
□	**Auditd** data retention is configured	*Section 4.2.2, “Configure **`auditd`** data retention”*
□	Infosec's **`audit.rules`** are installed	*Section 4.2.3, “Configure **`auditd`** rules”*
□	**Audisp** sends audit logs to syslog	*Section 4.2.4, “Send audit logs to syslog”*
□	**Sudo** is configured correctly	*Section 4.3, “Audit access with **sudo**”*
□	**Crond** and **atd** are restricted	*Section 5.3, “**`Cron`** and **`At`** daemons”*
□	**Ntpd** is configured	*Section 5.4, “Network Time Protocol Daemon (**`ntpd`**)”*
□	**Postfix** is installed and configured	*Section 5.5, “Mail Transfer Agent”*
□	**Sendmail** is removed	*Section 5.5, “Mail Transfer Agent”*
□	Root alias is set up	*Section 5.5.3, “Configure root alias”*
□	**Psacct** service is enabled	*Section 5.6, “Process accounting daemon (**`psacct`**)”*
□	**OpenSSH** is configured	*Section 5.7, “Configuring OpenSSH (**`sshd`**)”*
□	McGill Infrastructure services are installed (if applicable)	*Section 5.8, “McGill infrastructure services”*
□	All unnecessary services are disabled	*Section A.2, “Services to disable checklist”*

Table A.2. Post-installation checklist

A.2. Services to disable checklist

	Service	Caveats	Package
□	anacron		anacron
□	apmd		apmd
□	autofs	unless netfs automounting is used	autofs
□	avahi-daemon		*(req. by cups)*
□	bluetooth		bluez-utils
□	cups	unless printing is required	*(req. by LSB)*
□	firstboot		firstboot-tui
□	gpm	unless mouse is used at the console	gpm
□	haldaemon		*(req. by RHN)*
□	hidd		bluez-utils
□	kudzu		kudzu
□	lvm2-monitor	unless LVM2 mirroring is used	*(do not remove)*
□	netfs	unless NFS is used	*(do not remove)*
□	nfslock	unless NFS is used	nfs-utils
□	mcstrans	unless MLS SELinux policy is used	*(do not remove)*
□	mdmonitor	unless software RAID is used	mdadm
□	microcode_ctl	unless it's a 32-bit system	microcode_ctl
□	pcscd		pcsc-lite
□	portmap	unless NFS/NIS is used	portmap
□	readahead_early	unless you reboot a lot	readahead
□	readahead_later	unless you reboot a lot	readahead
□	rpcgssd	unless NFS is used	nfs-utils
□	rpcidmapd	unless NFS is used	nfs-utils
□	setroubleshoot		setroubleshoot
□	xfs	unless X11 applications are used	xorg-x11-xfs

Table A.3. Disabled services checklist

For more information on services that should be left enabled, see *Section 5.1.2, “Guidance on Default Services”*.

Important

It is recommended that you reboot after enabling or disabling services to make sure that all services are starting as expected on system boot.

Appendix B. Revision history

Revision 1.2 **August 26, 2010** **Konstantin Ryabitsev**
`konstantin.ryabitsev@mcgill.ca`

Added *Section 2.2, “Modify the system login banners”*.
Added wording that CIO/Legal Services approval will be required for keystroke auditing (*Section 4.2.7, “Using **`pam_tty_audit`** to log all keystrokes”*).

Revision 1.1 **August 24, 2010** **Konstantin Ryabitsev**
`konstantin.ryabitsev@mcgill.ca`

Added *Section 3.4, “TCP Wrapper”*.
Adjusted "VAS" naming to reflect the purchase by Quest.
Added a note on NTP and VAS interaction.
Added another sentence to advise removing the software entirely in addition to disabling the service.
Adjusted *Section A.2, “Services to disable checklist”* to list the packages that are safe and not safe to remove.
Remove "draft" designation.

Revision 1.0 **May 28, 2010** **Konstantin Ryabitsev**
`konstantin.ryabitsev@mcgill.ca`

Initial draft revision

Index